LATTICE: Democratize High-Fidelity 3D Generation at Scale Supplementary Material

Zeqiang Lai[1,2⋆] , Yunfei Zhao[2⋆] , Zibo Zhao[2] , Haolin Liu[2]
Qingxiang Lin[2] , Jingwei Huang[2] , Chunchao Guo[2‡] , Xiangyu Yue[1‡]
[1]MMLab, CUHK [2]Tencent Hunyuan
https://lattice3d.github.io

A. Discussions

A.1. Scaling Behavior on Different Architectures

We observe that 3D generation architectures differ substantially in their scaling behavior. Models without explicit localizable guidance (e.g., VecSet-based) scale much less effectively than those equipped with it, such as our proposed VoxSet architecture. Here, we provide a thorough analysis and accompanying demonstrations.

Model Scaling on Parameters. Our first observation is that VecSet models hardly benefit from an increased number of parameters. To investigate this, we compare three popular VecSet models of different sizes: Hunyuan3D-2-mini [3] (0.6B), Hunyuan3D-2 [9] (1.1B), and Hunyuan3D-2.1 [2] (3B). The visual comparison is shown in Fig. 1 (top row). Surprisingly, the three results are largely similar, with Hunyuan3D-2-mini (0.6B) even showing slightly better performance. This suggests that increasing model size may not play a decisive role for VecSet-based architectures. In contrast, we find that the proposed VoxSet models consistently benefit from increased model size. To demonstrate this, we train three models of different sizes (0.6B, 1.9B, and 4.5B). The generated results are shown in Fig. 1 (bottom row). It can be observed that as the number of parameters increases, the outputs exhibit more details, with sharper, smoother, and more regular structures.

Overall, these results demonstrate the effectiveness of the proposed VoxSet architecture, revealing that proper conditioning is a key factor in unlocking the benefits of model scaling. In other words, larger models are useful only when there is a clear correspondence between conditions and outputs; otherwise, increasing the number of parameters provides little advantage, as the model is more good at memorizing rather than abstracting and reasoning. From the data perspective, the difference in scaling behaviors could also be explained by the lack of large-scale 3D data. We need more data to cover mappings with higher degrees of freedom, such as those with fewer conditions.

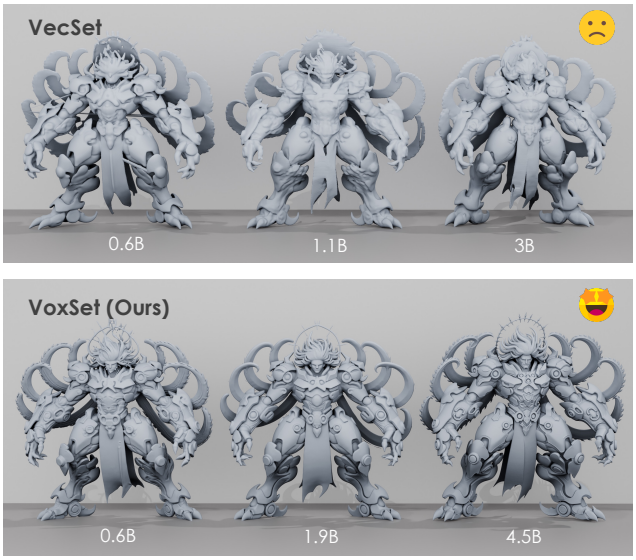

Figure 1. Illustration of the effect of model scaling (in parameters) on performance. VecSet models show limited improvement as parameters increase, whereas larger VoxSet models produce finer and more detailed results.

Test Time Scaling on Tokens. Previously, we demonstrated that our model exhibits a strong test-time scaling effect on shape tokens in Sections 1 and 3.2. Specifically, the model trained with a maximum token length of N can be directly evaluated using 2N, 3N, or even more tokens during inference—no additional training or configuration is required. Similar to model scaling, we here compare the test-time scaling behaviors of VecSet models and our VoxSet models, with results presented in Fig. 2. The VecSet model was trained with 4096 tokens, while our VoxSet model was trained with 6144 tokens. We evaluated their generation performance at token lengths of 1N, 2N, and 4N. As observed, the VecSet model benefits slightly from scaling tokens to 2N (e.g., improved round structures on the

⋆ Equal contribution. † Corresponding authors.

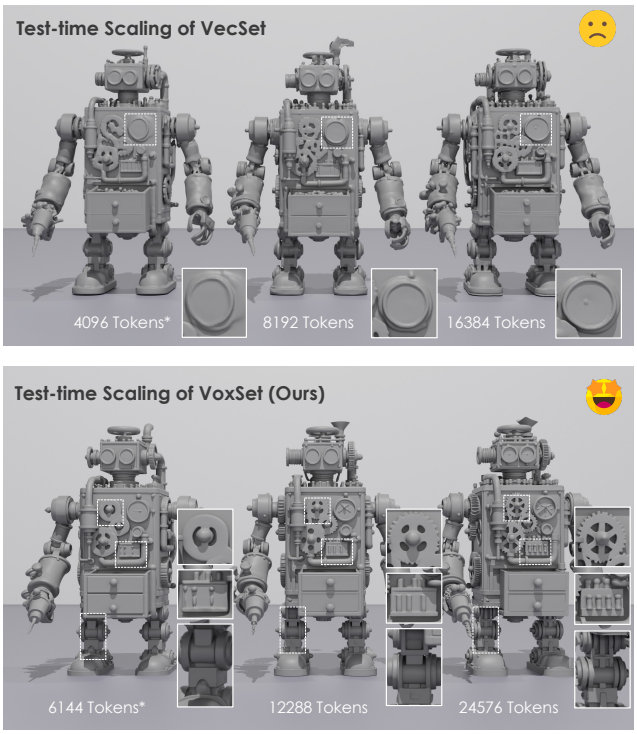

Figure 2. Illustration of the effect of test-time scaling (in shape tokens) on model performance. VecSet models exhibit limited gains as the number of tokens increases, showing early saturation. In contrast, VoxSet models consistently benefit from higher token counts, producing finer details and demonstrating stronger scaling capability. * indicates the token count used during training.

body), but further scaling yields negligible gains. In contrast, our VoxSet model consistently benefits from increased token counts: it shows improvements when scaling from N to 2N, and further gains when scaling from 2N to 4N. Notably, more tokens directly translate to richer details.

In general, the token scaling capability of both VecSet and VoxSet models originates from their ability to perform arbitrary-resolution autoencoding. During VAE training, we randomly sample queries across the entire 3D voxel grid or object surface—this process acts as a form of random dropout for full-token training. As a result, our VAE inadvertently acquires the ability to encode objects with any token length, even though this property was not an explicit design goal. The same principle applies to DiT training: the random selection of queries also endows DiT with test-time scaling capability. Notably, VoxSet exhibits a more pronounced scaling effect, which can be attributed to its stronger correspondence between query locations and content. During training, the transformer learns to model the relationship between spatial positions and the tokens that should be generated at those positions.

A.2. Representation Centric to Generation Centric

In this paper, our model challenges two common takes, which is largely from the view of representation, including *(1) Global vs Local*: "VecSet is global that is better for overall shape and sparse voxel is local which is more suitable for details", and *(2) Structural Sparse Grid vs Unstructural Set*: "the sparse voxel is structural which is better for editing and other downstream tasks, VecSet is unstructural for these tasks."

Localizable Code is All You Need. It is evident that we can combine the strengths of both approaches through the proposed VoxSet—a semi-structured set-based representation that encodes global information. Nevertheless, we wish to emphasize that while these two common perspectives hold true to some extent, critical nuances remain. From a generation standpoint, the key to better generation performance lies in localizability: specifically, strong guidance that is accessible during test time. Locality may offer advantages for compression efficiency and reconstruction quality, but it is not the primary factor for generation tasks. Regarding structure, sparse voxels are inherently structural, a property that benefits many tasks. However, this does not mean VecSet lacks structure. In fact, VecSet inherently contains structural information. Its only limitation is that this structure cannot be identified during test time. Thus, we propose VoxSet to circumvent this issue.

B. Implementation Details

Training Setup. To evaluate the scaling effect of the proposed architecture, we train several models of various sizes, including a medium model (0.6B parameters), an XL model (1.9B), and an XXL model (4.5B). Unlike CLAY [8], we do not adopt progressive model scaling; instead, all models are trained from scratch. Instead, we employ a multi-stage token scaling strategy. Within each stage, we use a constant learning rate with a linear warm-up, while gradually decreasing the base learning rate across stages from 1×10^{-4} to 1×10^{-6}. The batch size is maximized to fit GPU memory, reaching up to 2048 in our experiments. We utilize ZeRO-based optimizer, gradient, and parameter partitioning from DeepSpeed [7] to efficiently train large models on a distributed GPU cluster. All models are trained using the flow matching objective with a linear coupling plan, following the formulation in SiT [6]. Additionally, to enable classifier-free guidance [1], we randomly replace conditioning embeddings with zero embeddings at a probability of 10% during training.

Data Preparation. We utilize a subset of the dataset from HY3D-2.0, which includes roughly 1100k objects. Over 70% are filtered from Objaverse-XL, and the rest are from other trivial sources. Our data processing pipeline mainly includes three steps, (1) data filtering; (2) water-

tighting; (3) point-cloud sampling and SDF extraction. We apply extensive data filtering to improve the quality of the dataset, which includes removing AI-generated assets, scanned assets, extreme complex scenes, and assets with plane. We randomly sample millions of point cloud during the training, and split them into chunks to accelerate data loading.

C. Post-Training

High-Quality Finetuning. We introduce an additional finetuning stage with very high-quality data, which helps in improving details generation. All parameters in DiT are updated with a small learning rate. We filter the high-quality data by a combination of criterions including the number of faces, the number of sharp edges, and reconstruction quality, resulting in roughly 15k samples.

Model Acceleration. We adopt FlashVDM [3] to accelerate the geometry VAE decoding. For diffusion sampling, benifit from strong structure guidance from RoPE, we find that our models are inherent few-step generators. Nevertheless, we still perform guidance distillation and step distillation to further reduce the sampling cost.

D. User Study Setting

To evaluate the perceptual quality of the generated results, we conducted a large-scale user study consisting of approximately 500 questions. For each question, three independent participants were asked to rank all the presented results according to their visual quality or fidelity. Each question displayed outputs from all compared methods in a randomized order to ensure fairness. Finally, we aggregated the rankings across all participants and questions to compute the winning rate of each method, which reflects its overall preference by human evaluators.

D.1. More VAE Evaluation

Clarification on VAE Performance. SparseFlex's underperformance (metrics) stemmed from the 1/1536 resolution testing, which caused significant mesh fragmentation and holes in thin structures. Therefore, we re-evaluated at 1/512 thickness.

Resolution	TripoSF	Ours
1/1536	90.94	**98.53**
1/512	97.25	98.46

Table 1. F1 score between SparseFlex and our method.

Visual Comparison. We observe that our reconstruction outperforms all prior vecset-based methods and is competitive with sparse-voxel approaches, although it does not surpass them. As illustrated in Fig. 3, performance is generally comparable in most cases (see the first row). However, our method struggles with very small faces, while it performs particularly well on more challenging ("hard") facial cases. Overall, the primary advantage of our VAE lies in generation (DiT training) rather than reconstruction.

Figure 3. Visual comparison of reconstruction results.

D.2. Limitations and Future Work

While our method achieves strong results, there remains room for improvement in handling high-resolution visual inputs more effectively. Future work may explore multi-scale or adaptive patching schemes for more effective high-resolution image conditioning [4, 5]. Besides, it would be interested and valuable to explore the way to compress irregular data via VoxSet, which is typically awkward for sparse grid methods.

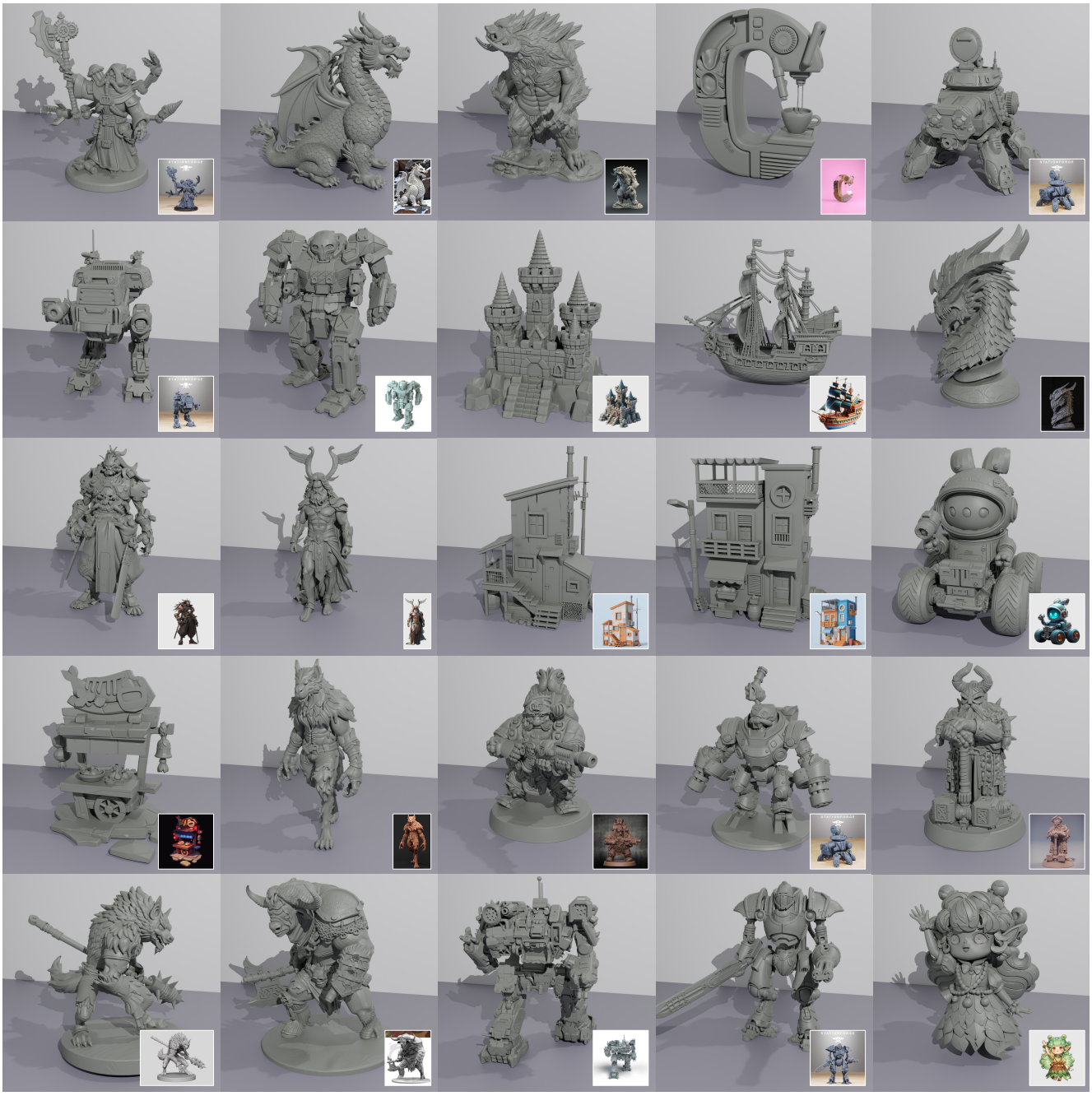

Figure 4. More visual results for image-to-geometry generation of LATTICE.

E. More Results

Fig. 4 shows additional results of our model. All examples are presented without cherry-picking.

References

[1] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *arXiv preprint arXiv:2207.12598*, 2022. 2

[2] Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, et al. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025. 1

[3] Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Fuyun Wang, Huiwen Shi, Xianghui Yang, Qinxiang Lin, Jinwei Huang, Yuhong Liu, Jie Jiang, Chunchao Guo, and Xiangyu Yue. Unleashing vecset diffusion model for fast shape gener-

ation, 2025. 1, 3

[4] Wenzhuo Liu, Fei Zhu, Shijie Ma, and Cheng-Lin Liu. Mspe: multi-scale patch embedding prompts vision transformers to any resolution. *Advances in Neural Information Processing Systems*, 37:29191–29212, 2024. 3

[5] Wenzhuo Liu, Weijie Yin, Fei Zhu, Shijie Ma, Haiyang Guo, Xiao-Hui Li, Cheng-Lin Liu, et al. One patch doesn't fit all: Adaptive patching for native-resolution multimodal large language models. In *International Conference on Learning Representations*, 2026. 3

[6] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024. 2

[7] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020. 2

[8] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43 (4):1–20, 2024. 2

[9] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025. 1